Reflections on 35 Years of Computing at Los Alamos

David W. Forslund

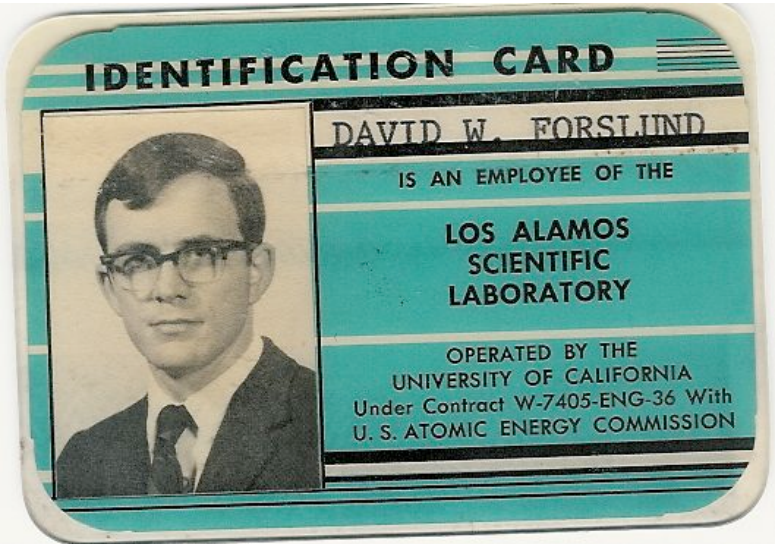
Laboratory Fellow

June 24, 2004

A View of Computing at Los Alamos

- Or “How I failed to avoid getting into trouble at Los Alamos”
- Or “How do we deal with Moore’s Law” (and Moore’s Law corollaries)
- Or “A journey with Seymour Cray in the early years”
- A little history
- Some philosophy
- More anecdotes and “defining moments”

Who on earth is this guy?



The Beginnings

- Came here directly from grad school (1969)with no intent to concentrate on computing (theoretical astrophysics/plasma physics)
- Conrad Longmire hired me; Norris Bradbury was Director
- C Division formed out of T-Division before I arrived.
- CDC 6600 was the workhorse computing platform
- As postdoc(!): 2 weeks of intro (Sam Glasstone):
  - Nuclear physics for everyone
  - Basics of bomb design for technical staff
  - Toured every experimental Lab facility

Some Philosophy

- "*Where there is no vision, the people perish,*" Proverbs
- Make mistakes faster than most people
- Remember the mistakes (less true today, unfortunately)
- The Lab should help the nation avoid "technological surprise"
  - National defense is a very broad area
- Let's stand on the shoulders of giants

System evolution

- CDC 6600
- CDC 7600
- Cray 1, X-MP, Y-MP
- Unix
- Object Oriented Computing
- High Performance Computing in the Commercial world
- Where is this going?

CDC 6600

- 100ns clock 128K (60bit words) of memory
- SCOPE operating system
- Extended Core Memory (ECM) 2Mwords
- Floating point parallel processing (couple Mflops)
- First experience with interactive computing
  - Teletype console (110baud) (Why did they call it the Silent 700?)
  - Punched tape (for restarts?!)

CDC 7600

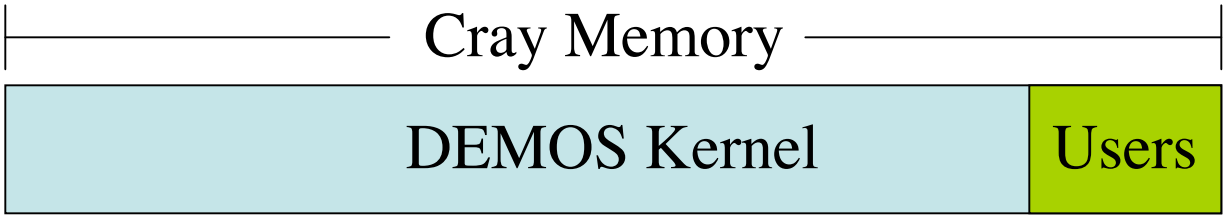
- **1969-70**
- **Hardware**
  - 27.5ns (36Mhz) clock
  - 36Mflops peak through pipelining
  - 65K words of memory (SCM)
  - LCM (Large Core Memory)
  - Had to deal with NUMA architecture then
- **Software**
  - CROS (Chili Ridge Operating System) LASL developed
  - Interactive computing through punch cards (Cafeteria)
  - High performance IO required for real applications
    - Bob Mitchell
- **Applications**
  - HANE work with Carl Shonk and Dick Morse
  - 1D/2D particle simulation codes
- **Experiences in India** (IBM 360 & BESM-6)
- **Transition to LTSS** (do you remember the @sorry?)

Cray-1

- 1976
- LASL rejected the CDC Star 100 and purchased Serial #1 Cray 1
  - Initially no operating system and then BOS
- 80MHz, 100Mflops peak, 8 Mbytes memory, 8 vector registers
- Vectorizing compilers were the big deal
- Serial performance could easily dominate the code, however.
- 2D WAVE code
- Parallel I/O critical
- Automated runs (when CFS came on line in 1979)

Cray/DEMOS/ALAMOS

- LASL brought Forest Baskett from Stanford to lead an effort in building a modern operating system for the Cray. (follow on to CROS)
- Modular with message passing (based on MODEL)
- Very successful research project (M. Powell "*A fast file system for Unix*")
- Not very successful as production system



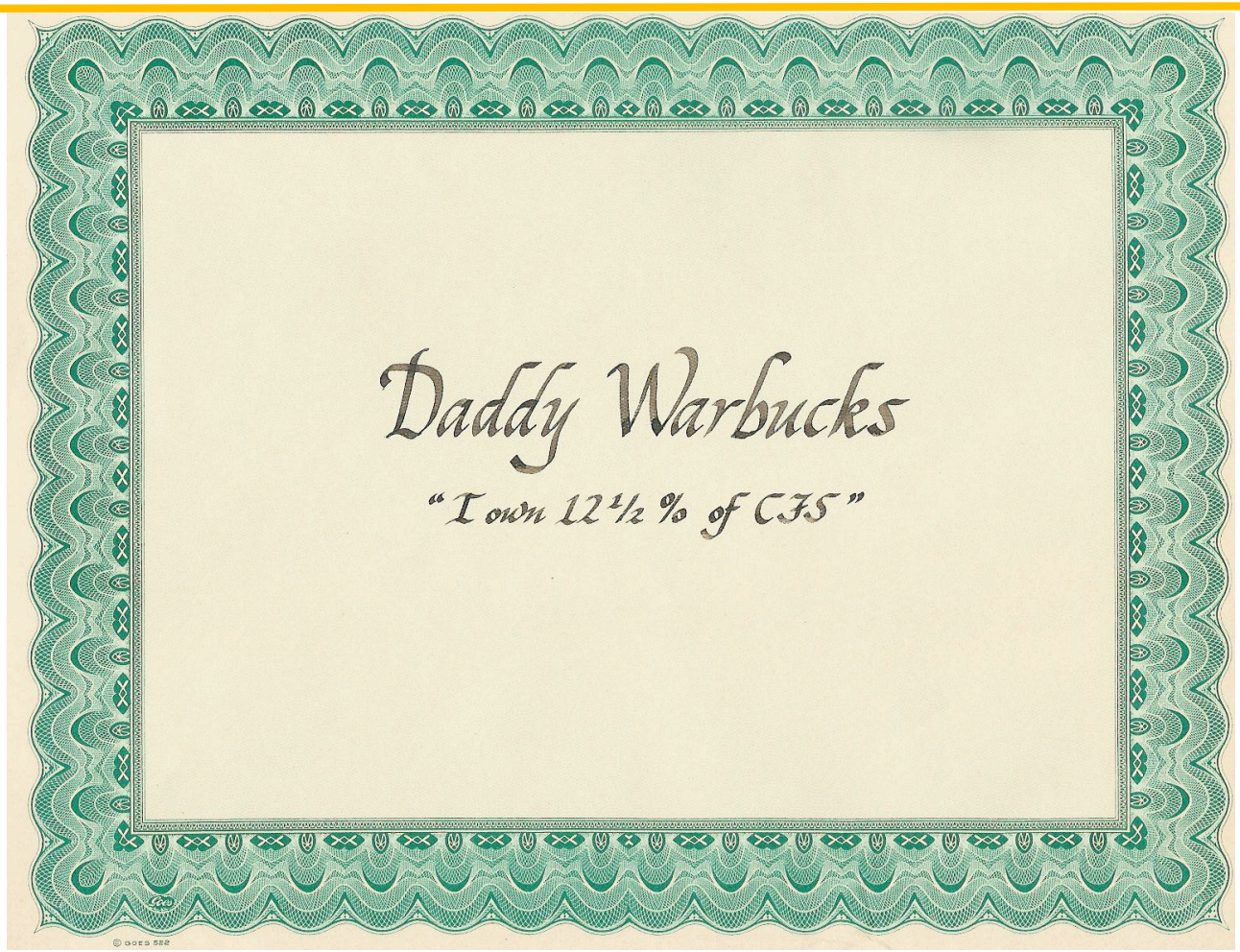
Cray/CTSS transition

- ALAMOS/DEMOS had severe performance problems particularly in the area of I/O and couldn't provide a timesharing environment
- Events surrounding the formation of X-Division
- CTSS was adopted in a dispute resolved at the Director's level
- Experiences with CTSS
- Ultimate Cray migrated to UNICOS

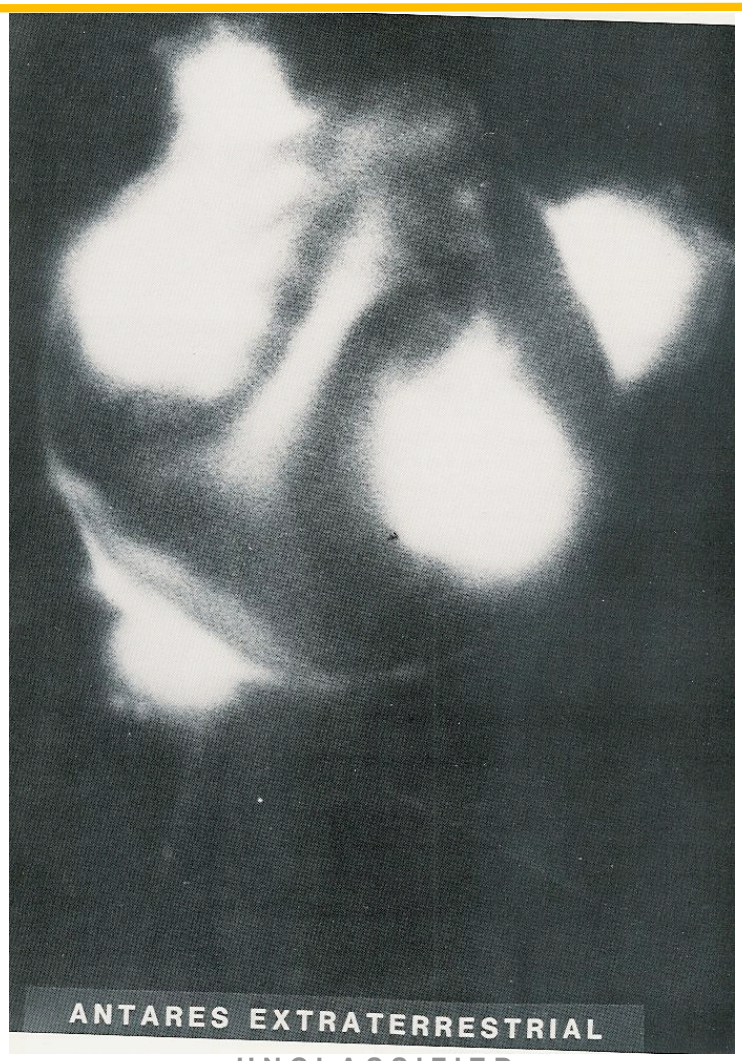
Cray X-MP performance

- Larry Rudsinski handed coded our WAVE kernel which ran almost unchanged for 18 years of the entire range of Cray 1-> Y/MP hardware.
  - Used vector registers as cache
  - Kernel exceeded compiler for this entire time by more than a factor of 2.
- Parallel I/O critical for performance
  - 4 channel I/O with polling required
- Solved major problems associated with ICF
  - Developed unusual ICF targets including ion accelerators and microwave emitters
  - Also landmark paper in laser beat-wave acceleration

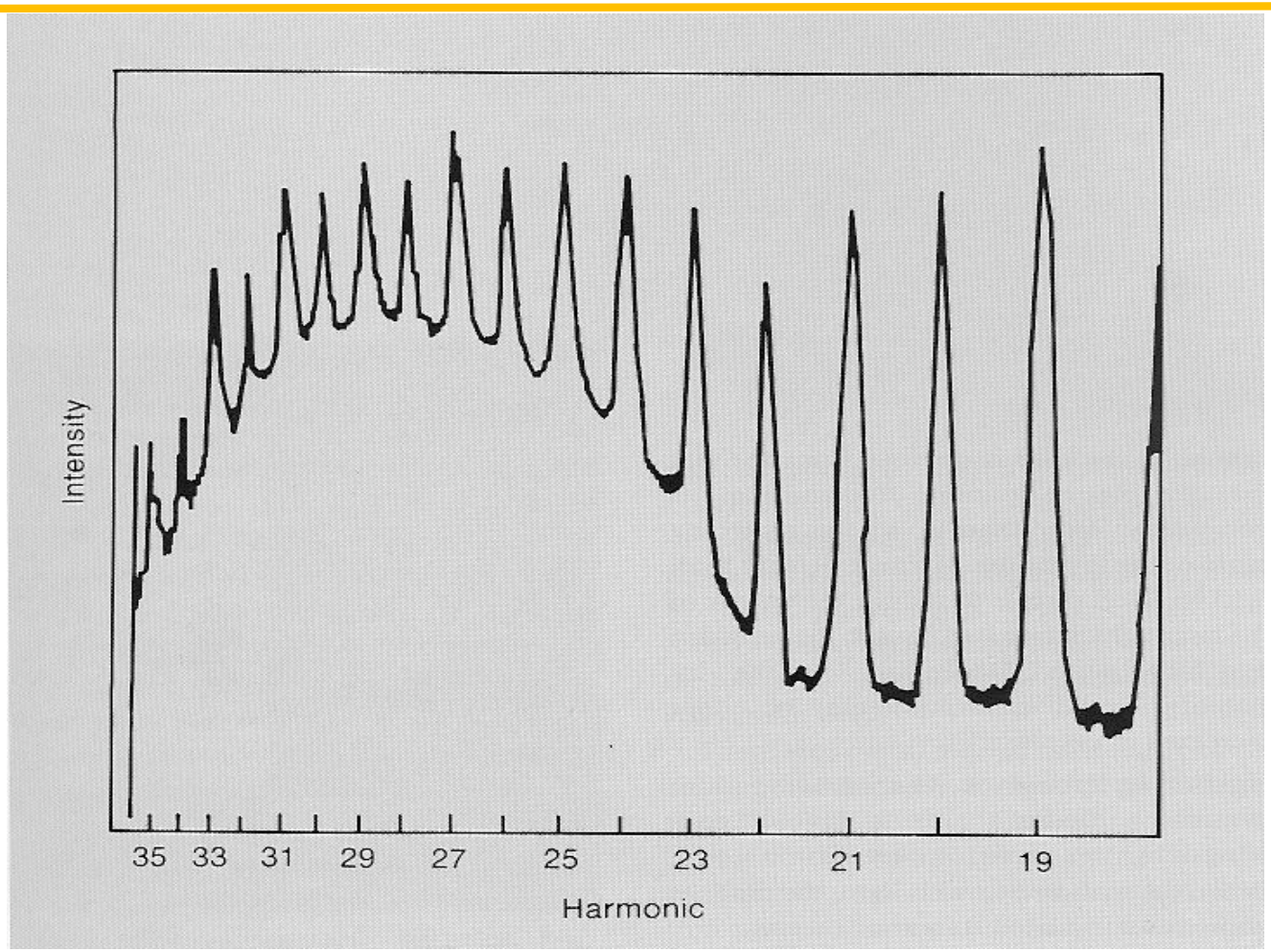
CFS really worked!



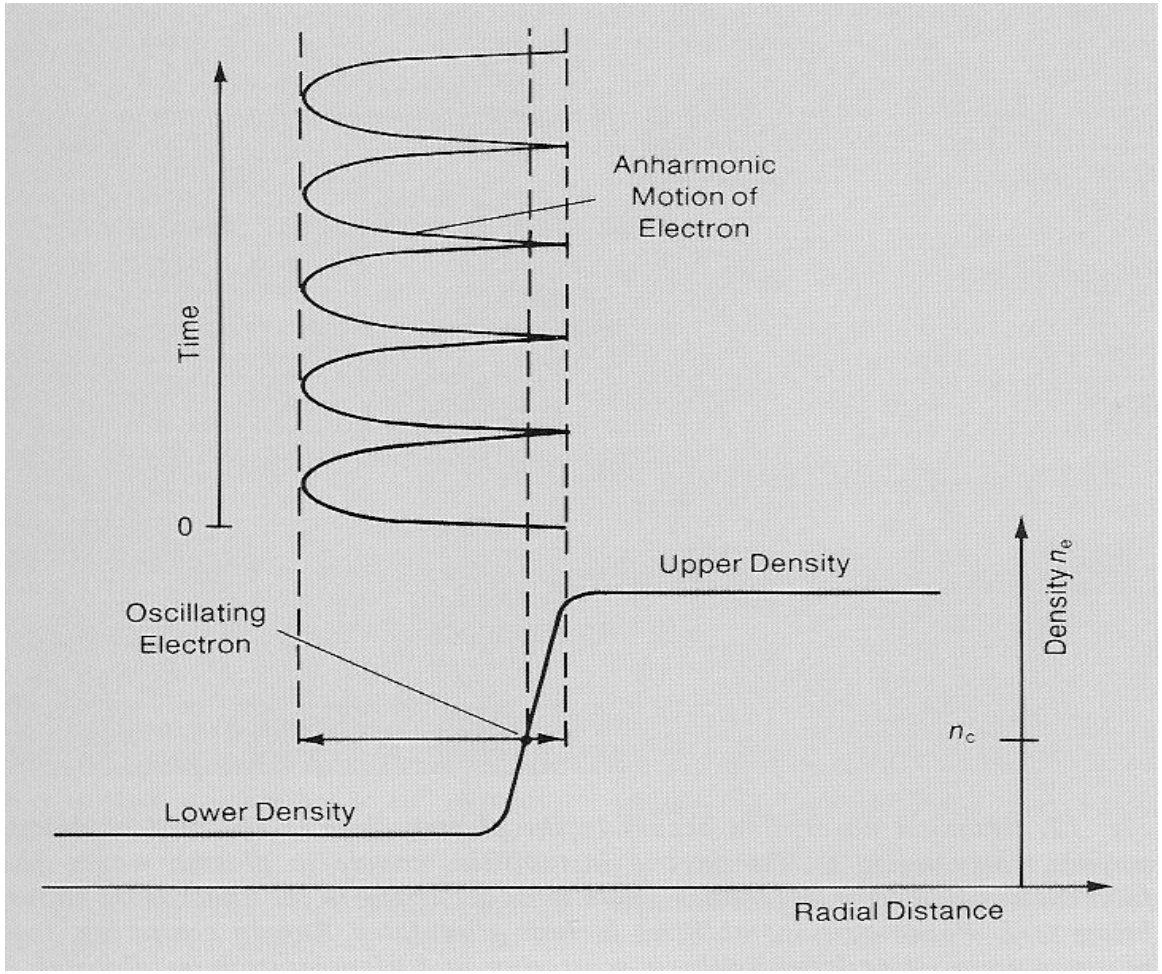
CO_2 Lasers Produce Enormous Magnetic Fields



Intense CO_2 Laser Light produces UV harmonics



Intense CO_2 Laser Light creates a highly nonlinear interaction



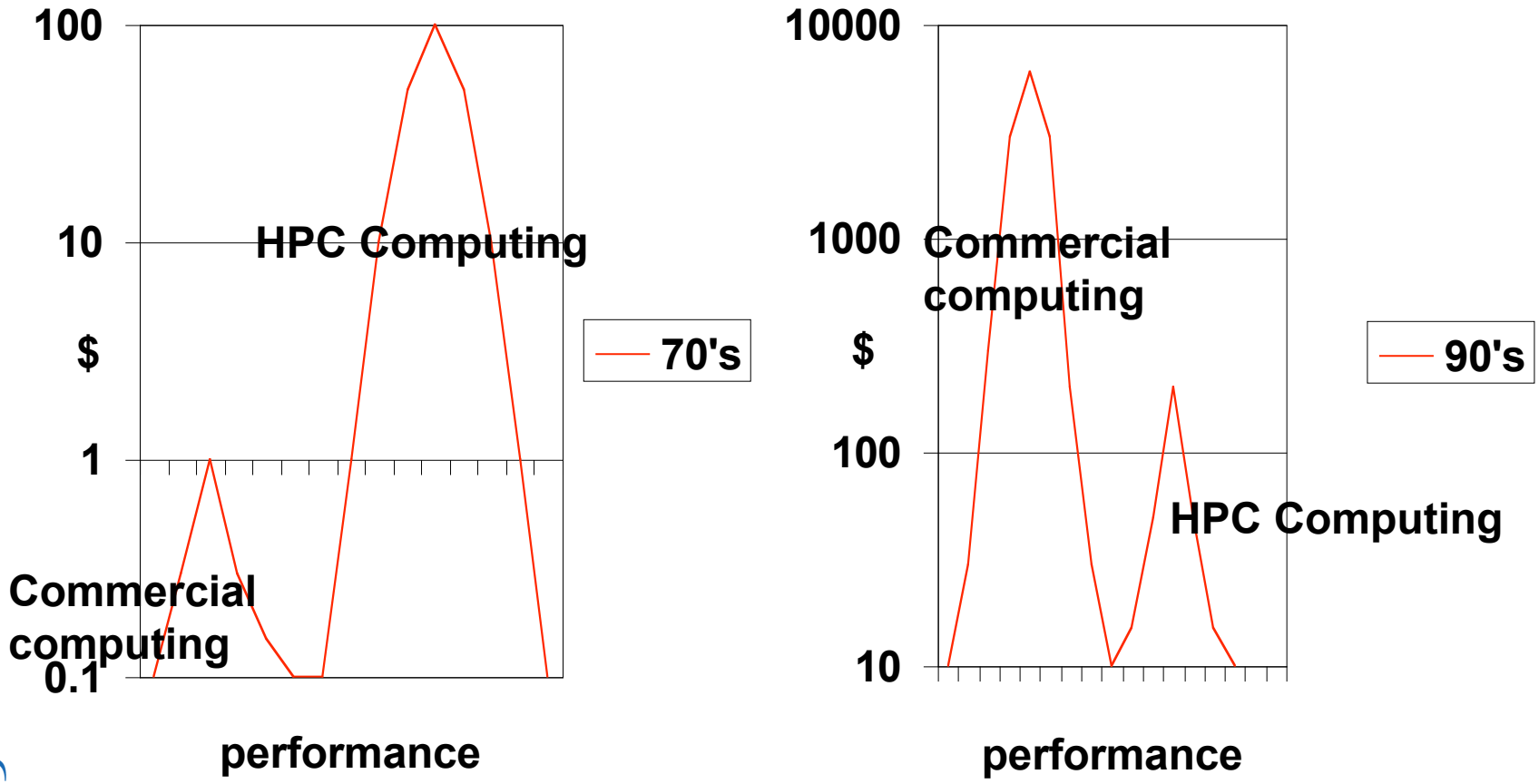
BUT the world was changing!

- Moore's law in micros was catching up with mainframes
- Unix on the desktop in the 80's (Unix wars)
- Sun workstation (Sun 2/120, 1Mbyte memory, 40Mbyte hard drive: only $26K)
- Mid 80's Sun 3/50 ran WAVE faster in wall-clock time than Cray 1.
- VAXes were disappearing (Did DEC ever "get it"?)
- Massively parallel computing coming onto the scene
- Software tools "infinitely" more capable than on the Cray
  - Networked computing was coming into vogue
  - NeWS demonstrated on the Cray between LANL and LLNL
- Object oriented design becoming popular

Consequence of Moore's Law

Source: Intel WSTS, 12/02 Approx

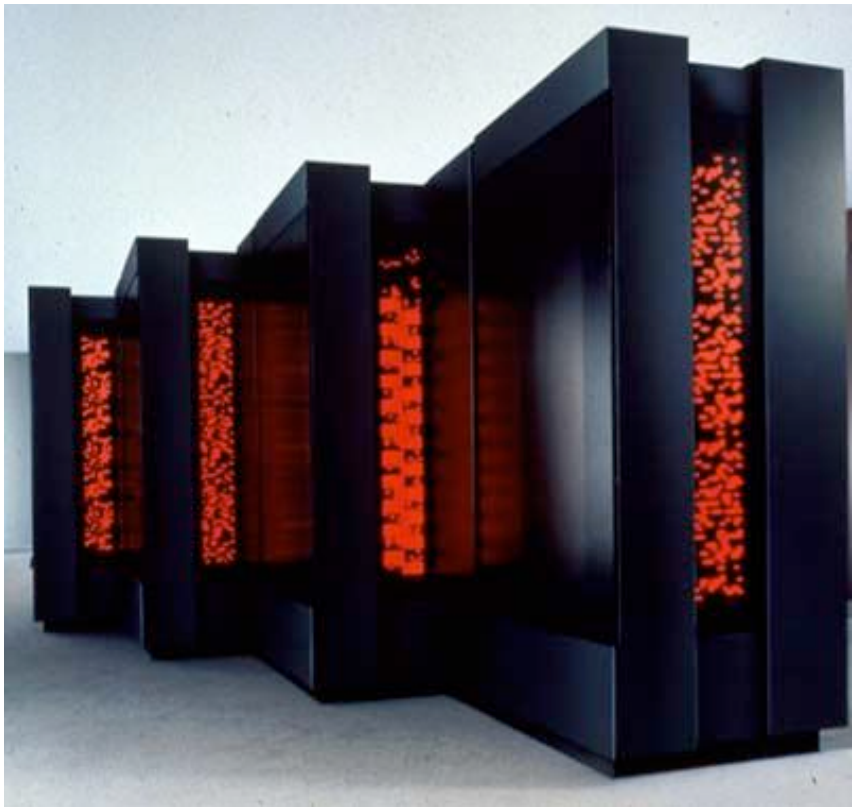
Shift in computing $



The ACL's new building

CM-2/CM-5

- SIMD/MIMD Computing model made parallel computing easy, but not for all problems.
- CM-5 was a wonderful demonstration of the power of parallel computing
- Thinking Machines told us we owned 20% of all their CPU's.
  - This was bad news.
- HPC was harder than many people anticipated.
  - Working with industry had mixed success
  - Why?



Some Comments on Cluster computing

- My view is that the software hasn't been able to track the changes in parallel/ distributed computing as it should.
- Need to leverage the millions of "programmers" out there.
- MPI is not the paradigm for scalable computing from a software investment perspective.

Platform independent HPC

- Need software that is independent of platform and OS.
- Computing must become a commodity not an art or craft.
  - Danny Hillis had a vision before Grid computing
- “We” chose to work with industry to leverage what they are doing and ride Moore’s law and the industry marketplace

Sunrise Project

- 1993 began the Sunrise LDRD project to demonstrate what we thought the new “NII” should look like
- Very successful, but not understood by industry.
- Since then we have worked with standards bodies to get these ideas incorporated into industry standards
  - Enable us to leverage commercial products more effectively
  - Solve broader range of society problems
- OpenEMed and B-SAFER Medical Surveillance are outcomes of this effort

My Current approach

- Programming in Java and riding its performance curve which now can provide equal or better performance than C
- Platform and language independent distributed computing
- Rapid evolution of software today may be its greatest enemy
- Can we really write code that can adapt to new hardware efficiently?
- Can we make parallel computing “disappear”?
- What are the important problems to be solved?

Commercial/Lab computing convergence

- E.g., need-to-know access is needed in healthcare (HIPAA) and finance
  - Standard for managing need-to-know has been established
- Open Standards/ Open Source/Open Components
- Commodity parallel computing is occurring, sometimes not the way we think
- Managing information is becoming increasingly important including “data-mining”. Needed by everyone.
  - What ever happened to my 1/8 of CFS?
- Transitioning software from one context to another may be the biggest obstacle to overcome
- What lies ahead?
- Despite my occasional grumblings, my professional success has been limited only by me and not by the Laboratory. LANL has provided me with essentially unlimited opportunities to succeed professionally, and for that I’m very grateful.